An Exact Method for a Green Vehicle Routing Problem with Traffic Congestion

Hongyuan Luo, Mahjoub Dridi, Olivier Grunder

An Exact Method for a Green Vehicle Routing Problem with Traffic Congestion*

Hongyuan Luo, Mahjoub Dridi, and Olivier Grunder

Nanomedicine Lab, Univ. Bourgogne Franche-Comté, UTBM, 90010 Belfort, France
{hongyuan.luo,mahjoub.dridi,olivier.grunder}@utbm.fr

Abstract. This paper addresses a time-dependent green vehicle routing problem (TDGVRP) with the consideration of traffic congestion. In this work, the objective is to optimize the vehicle routing plan, with the goal of reducing carbon emissions, which has linear relationship with fuel consumption of vehicles. To deal with traffic congestion, travel time are considered to be time-dependent. We propose a branch-and-price (BAP) algorithm to precisely solve this problem. A tailored labeling algorithm is designed for solving the pricing sub-problem. Computational experiments demonstrate the effectiveness of the proposed BAP algorithm.

Keywords: Vehicle routing problem · Time-dependent · Carbon emissions · Branch-and-price.

1 Introduction

The vehicle routing problem (VRP) is a practical and concerned issue in a wide range of application systems, including logistics, transportation, distribution, home health care, and supply chains [1]. Recently, the growing awareness of environmental concerns such as global warming and urban air pollution has led to increased efforts to protect the environment [2]. Many companies consider environmental-friendly operations throughout their supply chains. The motivation of being more environmental conscious is not only about legal constraints, but it also reduces costs and attracts customers who prefer green operations [3]. Therefore, some researchers have studied the green VRP (GVRP).

Nowadays, traffic congestion has become another common problem. Based on this limitation, [4] proposed the time-dependent VRP (TDVRP) in which the travel time changes with the departure time. In this paper, we focus on the time-dependent green vehicle routing problem (TDGVRP) with the time-varying vehicle speed and time windows. The cost of fuel consumption is one of the most significant part of the operation costs in logistics transportation. In the sense of the TDGVRP, we strive to minimize carbon emissions, which has linear relationship with fuel consumption, as the objective to optimize the

* The first author would thank the China Scholarship Council for the financial support gratefully. (contract No. 201801810122).

vehicle routing with the time-varying vehicle speed. An exact branch-and-price (BAP) algorithm is developed to solve the TDGVRP.

The rest of this paper is organized as follows. Section 2 introduces the problem and formulation. Section 3 develops an exact BAP approach to solve the problem. The computational experiments are described in Section 4. Section 5 concludes the paper.

2 Problem description and formulation

2.1 Problem description

The studied TDGVRP addresses a key aspect regarding the impact of traffic congestion in terms of the feasibility, fuel consumption and carbon emissions of a route. Let $G = (N, A)$ be a directed graph with a set of vertices $N = \{0, 1, ..., n, n+1\}$, in which $N_c = \{1, ..., n\}$ represent customers, and 0 and $n+1$ represent the origin and the destination depot, respectively. The distance between vertex i and vertex j is denoted by d_{ij}. Each vertex $i \in N$ has a demand q_i, a service time s_i, and an associated hard time window $[a_i, b_i]$. Each vehicle has to wait until the earliest time a_i if it arrives at vertex i before time a_i, meanwhile the vehicle is prohibited to arrive after time b_i. For vertices $i \in \{0, n+1\}$, it is assumed that $q_i = s_i = a_i = 0$ and $b_i = T$. Thus there is a fixed planning horizon $[0, T]$ in which vehicles are allowed to move along the route. A set of unlimited fleet of homogeneous vehicles with a limited capacity Q are employed to service the customers.

The impact of traffic congestion is captured by considering that the time required to travel from vertex i to vertex j, when departing from vertex i at time t, is given by a travel time function that is continuous, piecewise linear, and satisfies the FIFO property (i.e., a later departure always leads to a later arrival and thus overtaking will not happen). The traffic congestion at peak hours influences vehicle speed directly, thus affecting the travel time and fuel consumption of the vehicles. Under this setting, the cost minimizing the TDGVRP consists in finding a set of feasible routes minimizing the total carbon emissions, which has linear relationship with fuel consumption of vehicles.

2.2 Modeling time-dependent travel time and carbon emissions

The time dependence of the studied TDGVRP is based on the interaction between vehicle speed profiles determined by the traffic condition and travel time function [2]. According to the survey and literature, there are two peak traffic jam periods in a day, respectively are from 7 am to 9 am and from 5 pm to 7 pm. Fig. 1 (a) draws an average vehicle speed profile of departure time from 7 am to 7 pm. Fig. 1 (b) draws the corresponding travel time function for an arc of the distance 50 km. By using those stepwise speed functions, the FIFO property holds for each arc in the graph G. In this paper, let $\tau_{ij}\left(t_i^k\right)$ represent the travel time from vertex i to vertex j when vehicle k departs from vertex i at time t_i^k and so $\tau_{ij}(t)$ is a travel time function about the departure time t.

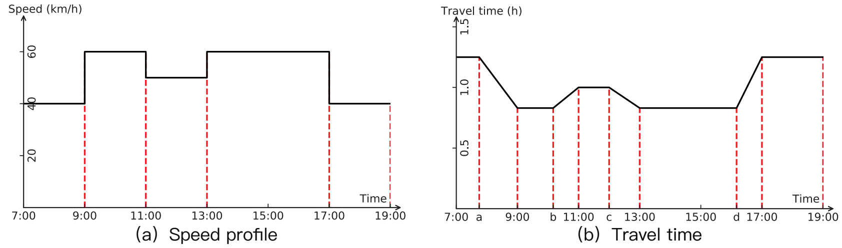

Fig. 1. Piecewise linear travel time function derived from stepwise speed function for an arc of length 50 km.

For each arc $(i, j) \in A$, there are several time periods of the corresponding travel time function $\tau_{ij}\left(t_i^k\right)$. In this paper, we define T_{ij} as the set of time periods. For example, if Fig. 1 (b) is travel time function of an arc (i, j), then there will be 9 time periods in travel time function. We denote m as the index of the time periods, that is $T_{ij}^m \in T_{ij}$ for $m = 0, 1, ..., |T_{ij}| - 1$, which T_{ij}^m is defined by two continuous time breakpoints w_m, w_{m+1}, namely $T_{ij}^m = [w_m, w_{m+1}]$. Due to the travel time function $\tau_{ij}\left(t_i^k\right)$ is linear in each time period $T_{ij}^m \in T_{ij}$, it is easy to get the linear function expression by calculating the slope θ_m and intercept η_m, namely $\tau_{ij}\left(t_i^k\right) = \theta_m \times t_i^k + \eta_m$.

For a path $p = (v_0, v_1, ..., v_k)$ which $v_0 = 0$ is the original depot and v_i for $0 \le i \le k$ is the the vertex at position i in the path p, the earliest time when the departure time at vertex v_0 is t and the service at v_i is completed is represented by the ready time function $\delta_{v_i}^p(t)$. The ready time function is nondecreasing in the domain t, and can be calculated for each vertex in the path p as follows:

$$\delta_{v_i}^p(t) = \begin{cases} t & \text{if } i = 0, \\ \max\{a_{v_i} + s_{v_i}, \delta_{v_{i-1}}^p(t) + \tau_{v_{i-1}v_i}\left(\delta_{v_{i-1}}^p(t)\right) + s_{v_i}\} & \text{otherwise.} \end{cases} \quad (1)$$

The ready time function is also piecewise linear, and similarly we can use the breakpoints, and the the boundary values of the time window at vertex v_i to represent the ready time function.

The research addressed in this paper aims to minimize the carbon emissions by optimizing vehicle speeds and routes. Carbon emissions is used to provide an estimate of the exhaust generated by vehicles. To measure carbon emissions, the speed-emission coefficients should be applied first to estimate the fuel consumption. "Road Vehicle Emission Factors 2009" has reported a database of vehicle speed-emission factors for fuel consumption [5]. The general format of the vehicle fuel consumption function is presented as:

$$FC(v) = k\left(a + bv + cv^2 + dv^3 + ev^4 + fv^5 + gv^6\right)/v \quad (2)$$

where v is the speed in km/h, $FC(v)$ is the fuel consumption in $l/100$ km, and k, a, b, c, d, e, f, g are different coefficients for estimating the fuel consumption. In this paper, we adopt the coefficients in [6], and the corresponding coefficients for each $l/100km$ fuel consumption are 0.037, 12690, 16.56, 86.87, -3.55, 0.06146, -0.0004773, and 0.000001385, respectively. The conversion factor of carbon emissions from fuel consumption is 3.1787 kg carbon per liter fuel consumed. Therefore, the formula of carbon emissions is $CE\left(v\right) = 3.1787 \times FC\left(v\right)$.

As for an arc $(i,j) \in A$, we define $f_{ij}\left(t_i^k\right)$ as the carbon emissions function when vehicle k departs from vertex i at time t_i^k. We denote v_m as the speed of time period $T_{ij}^m \in T_{ij}$ in travel time function. So there are same speeds in consecutive time periods, such as in Fig. 1 (b), $v_0 = v_1 = 40$ km/h. Based on travel time function formula $\tau_{ij}\left(t_i^k\right) = \theta_m \times t_i^k + \eta_m$, if the slope θ_m of the time period to which the departure time t_i^k belongs is 0, namely $\theta_m = 0$, then there will be only one speed v_m used in the journey between vertex i and vertex j. Therefore $f_{ij}\left(t_i^k\right) = CE\left(v_m\right) \times d_{ij}/100$. And if the slope θ_m of the time period to which the departure time t_i^k belongs is not 0, namely $\theta_m \neq 0$, then there will be two speeds v_m, v_{m+1} used in the journey between vertex i and vertex j. Therefore

$$\begin{aligned} f_{ij}\left(t_i^k\right) = CE\left(v_m\right) \times v_m \times \left(w_{m+1} - t_i^k\right)/100 + \\ CE\left(v_{m+1}\right) \times v_{m+1} \times \left[t_i^k + \tau_{ij}\left(t_i^k\right) - w_{m+1}\right]/100. \end{aligned} \tag{3}$$

We analyze the formula 3 and find that the formula 3 is a linear function of t_i^k. Thus the carbon emissions function $f_{ij}\left(t_i^k\right)$ is also piecewise linear.

For a path $p = (v_0, v_1, ..., v_k)$ which $v_0 = 0$ is the original depot and v_i for $0 \leq i \leq k$ is the the vertex at position i in the path p, the earliest time when the departure time at vertex v_0 is t, we define $F_{v_i}^p\left(t\right)$ as the total carbon emissions function up to vertex v_i in path p when the service at v_i is completed with the ready time $\delta_{v_i}^p\left(t\right)$. As mentioned before, the ready time function $\delta_{v_i}^p\left(t\right)$ is also piecewise linear. Thus in path p, we can use the composite function to denote $f_{v_{i-1},v_i}\left(t_{v_{i-1}}^p\right)$ for $i = 1,2,...,k$, namely $f_{v_{i-1},v_i}\left(t_{v_{i-1}}^p\right) = f_{v_{i-1},v_i}\left(\delta_{v_{i-1}}^p\left(t\right)\right)$. Therefore, the total carbon emissions function is denoted as:

$$F_{v_i}^p\left(t\right) = \begin{cases} 0 & \text{if } i = 0, \\ F_{v_{i-1}}^p\left(t\right) + f_{v_{i-1},v_i}\left(\delta_{v_{i-1}}^p\left(t\right)\right) & \text{otherwise.} \end{cases} \tag{4}$$

where the addition operation between functions only counts the domain intersection. If the intersection is an empty set, then the function has no domain.

3 Branch-and-price algorithm

3.1 Set-partitioning formulation

We define Ω as the set of feasible paths. Let y_p be a binary variable deciding whether path p is included in the optimal solution or not, and let σ_{ip} be a binary variable that denotes the customer i is visited by the path p or not. We formulate the studied problem as a set partitioning formulation, which is presented as follows:

$$Z = \min \sum_{p \in \Omega} c_p y_p \tag{5a}$$

$$s.t. \quad \sum_{p \in \Omega} \sigma_{ip} y_p = 1, \ \forall i \in N_c \tag{5b}$$

$$y_p \in \{0, 1\}, \ \forall p \in \Omega \tag{5c}$$

where the objective function Z (5a) minimizes the carbon emissions of the chosen paths, constraint (5b) guarantees that each customer $i \in N_c$ is visited only once, and constraint (5c) ensures that the decision variables are binary.

We define the LP relaxation of the set-partitioning model as the master problem (MP). In the LP relation problem, the '=' in formula (5b) can be replaced by the '$\geq$', and the formula (5c) is replaced by $y_p \geq 0, \ \forall p \in \Omega$. We use column generation [7] to solve the MP with a small subset $\Omega' \subseteq \Omega$ of feasible paths. The MP with the subset Ω' is denoted as the restricted MP (RMP).

After the initialization step, new columns are added iteratively until the algorithm converges to the optimal (fractional) solution. At each iteration, the LP relaxation of the RMP is computed to obtain the dual variables π_i associated with constraints (5b) for $i \in N_c$. If a feasible route p with negative reduced cost $c_p - \sum_{i \in N_c} \pi_i \sigma_{ip}$ exists, then it is added to the RMP and the procedure is repeated. Otherwise, the current fractional solution is optimal.

3.2 Branching and node selection strategy

After column generation, a branching strategy on arcs is adopted in this paper. Let H_{ij} be the set of all columns that contain arc $(i, j) \in A, i, j \in N_c$. The sum of the flows on arc (i, j) is equal to $\sum_{p \in H_{ij}} y_p$. If there exists at least an arc (i, j) with fractional $\sum_{p \in H_{ij}} y_p$, then we branch on the value $\sum_{p \in H_{ij}} y_p$ which is the closest to the midpoint 0.5. Two new child nodes are generated accordingly by forcing arc (i, j) in one node and forbidding arc (i, j) in the other node. In the former case, all columns containing arcs (i, j') and (i', j) with $i' \neq i$ and $j' \neq j$ are deleted. In the latter case, columns using arc (i, j) have to be removed.

3.3 The pricing problem

The pricing sub-problem constructs a feasible route with a minimum reduced cost, using the dual values obtained from the LP solution of the RMP. If the constructed route has negative reduced cost, its corresponding column is added to the RMP. Otherwise, the LP procedure will be terminated with an optimal solution to the continuous relaxation of the MP. To sum up, at each iteration of the column generation algorithm, the pricing problem aims to find routes (columns) $r \in \Omega$ with negative reduced cost $\bar{c}_r$, if any exists. The pricing problem searches for the routes with a negative reduced cost, and its objective function is defined as $\min_{p \in \Omega'} \ \bar{c}_p = c_p - \sum_{i \in N_c} \pi_i \sigma_{ip}$, where $\bar{c}_p$ is the reduced cost of path p, and π_i is the dual variable associated with the formulation (5b). To solve the pricing problem, a forward labeling algorithm is developed.

The labeling algorithm The labeling algorithm generates implicitly an enumeration tree where each node is named as a label L and denotes a partial path starting from the original depot. Aiming to overcome the huge exponential growth, domination rules are used to reduce the number of labels. Then we present the forward labeling algorithm for the TDGVRP. In labeling algorithm, we start generating labels from the start depot o to its successors. For each label L, the notations are presented in Table 1.

Table 1. The notations of a label.

Notation	Interpretation
$p(L)$	The partial path of label L, and $p(L) = (o, ..., v)$.
$v(L)$	The last vertex visited on the partial path $p(L)$.
$S(L)$	The set of unreachable customers after visiting $v(L)$, and $S(L) \supseteq p(L)$.
$L^{-1}(L)$	The parent label from which L originates by extending it with $v(L)$.
$q(L)$	The total demand after servicing vertex $v(L)$ in path $p(L)$.
$\delta_L(t)$	The piecewise linear function that represents the ready time at $v(L)$ if vehicle departed at the origin depot at t and reached $v(L)$ through partial path $p(L)$.
$F_L(t)$	The piecewise linear function that represents total carbon emissions at $v(L)$ if the vehicle departed at the origin depot at t and reached $v(L)$ through path $p(L)$, namely $c_L(t)$.
$\pi(L)$	The cumulative value of dual variable associated with the formulation (5b) in path $p(L)$.

The labeling algorithm begins with the label $v(L) = 0$ that denotes the initial path $p(L) = (0)$, the algorithm iteratively processes each label L until no unprocessed labels remain. A label L' can only be extended to label L along an arc $(v(L'), j)$ when the extension is feasible with the constraints of time windows and capacity, namely $\delta_{L'}(t) + \tau_{v(L'),j} \leq b_j \quad \forall \quad j \in N \setminus S$, and $q(L') + q_j \leq Q \quad \forall \quad j \in N \setminus S$. If the extension along the arc $(v(L'), j)$ is feasible with the above conditions, then a new label L is created. The information of new label L is updated by using the following formulas: $p(L) = p(L') \cup \{j\}$, $v(L) = j$, $S(L) = p(L) \cup \{k \in N \vee \min\{\delta_{L'}(t) + \tau_{v(L'),k}\} > b_k \vee q(L') + q_k > Q\}$, $L^{-1}(L) = L'$, $q(L) = q(L') + q_j$, $\delta_L(t) = \max\{a_j + s_j, \delta_{L'}(t) + \tau_{v(L'),j} + s_j$, $F_L(t) = F_{L'}(t) + f_{v(L'),j}(\delta_{L'}(t))$, and $\pi(L) = \pi(L') + \pi_j$.

For a partial path $p(L)$, if the path $p(L)$ is feasible, then it will always be feasible with a departure time 0 from the origin depot o. In other words, if the path $p(L)$ is feasible, $dom(\delta_L)$ is always a time interval $[0, t_0]$ for $t_0 \geq 0$ with $dom(\delta_L)$ is the domain of δ_L.

After an extension to vertex j, if the last vertex in the new path $p(L)$ is $n+1$, that is $v(L) = n+1$, then the minimal cost c_L and the reduced cost $\bar{c}_L$ of the complete path associated with the label L are calculated by $c_L = \min_{t \in dom(F_L)}\{F_L(t)\}$, and $\bar{c}_L = \min_{t \in dom(F_L)}\{F_L(t)\} - \pi(L)$, where $dom(F_L)$ is the domain of the total carbon emissions function $F_L(t)$. Define $dom(\delta_L)$ and $img(\delta_L)$ be the domain and image of the ready time function $\delta_L(t)$, and let $dom(F_L)$ and $img(F_L)$ be the domain and image of the total carbon emissions function $F_L(t)$, respectively. Based on the definitions of $dom(\delta_L)$ and $dom(F_L)$, we can get that the domain of ready time function $\delta_L(t)$ is equal to the domain of total carbon emissions function $F_L(t)$, namely $dom(\delta_L) = dom(F_L)$.

In general, the labeling algorithm is similar to enumeration method and dynamic programming, and all possible extensions are handled and stored for each label. With the iteration of the labeling algorithm, the number of labels will exponentially increase. In order to overcome the exponential growth, dominance rules are used to reduce the number of labels enumerated. [8] define that three requisites must be met if Label L_2 is dominated by label L_1. These requirements are (1) $v(L_1) = v(L_2)$; (2) all of the feasible extensions $E(L_2)$ of the label L_2 to vertex $n+1$ must be the subset of label $E(L_1)$, namely $E(L_2) \subseteq E(L_1)$; (3) $\bar{c}_{L_1 \oplus L} \leq \bar{c}_{L_2 \oplus L}, \forall L \in E(L_2)$. Therefore, in the developed forward labeling algorithm, dominance rule is proposed as follows.

Proposition 1. *(Dominance rule) Label L_2 is globally dominated by label L_1 if (1) $v(L_1) = v(L_2)$, (2) $S(L_1) \subseteq S(L_2)$, (3) $dom(F_{L_2}) \subseteq dom(F_{L_1})$, (4) $\delta_{L_1}(t) \leq \delta_{L_2}(t), \forall\, t \in dom(\delta_{L_2})$, (5) $F_{L_1}(\arg\{\delta_{L_1} = \delta_{L_2}(t)\}) - \pi(L_1) \leq F_{L_2}(t) - \pi(L_2), \forall t \in dom(F_{L_2})$, and (6) $q(L_1) \leq q(L_2)$.*

4 Numerical experiments

In this section, we conduct the experiments to test the performance of the BAP algorithm. The algorithms are coded in C++. The RMPs are solved by CPLEX 12.10.0. Computational experiments are conducted on a PC with an Inter(R) Core(TM) i7-7700 CPU @3.60GHz × 8 and a 16 GB RAM, under a Linux system. Computation times are reported in seconds on this machine.

There are no similar problems in the existing researches, so we generate the test instances based on the classical Solomon VRPTW benchmark instances [9]. In this paper, we test the instances with 25 customers. In the basis of Solomon VRPTW instances, the rules of generating the test instances are as follows: (1) we set the start depot and the end depot as the same point; (2) the planning horizon was set as 12 h, therefore, the time window $[a_i, b_i]$ of vertex i in the Solomon instances was modified as $[a_i \times (12/b_0), b_i \times (12/b_0)]$; (3) the distance d_{ij} was not changed, but we set the unit as km; (4) the service time was set to 0.5h for all customers; (5) the other parameters were not changed.

Table 2. The results of BAP algorithm on instances with 25 customers.

Instance	Root LB	Root time	Best LB	UB	Time	Final Gap (%)	Nodes
C101	285.96	0.08	294.82	294.82	0.26	0	5
C102	250.66	0.62	260.94	260.94	1.94	0	9
C103	237.51	7201.7	238.94	238.94	7208.36	0	1
C104	208.88	7201.76	208.88	208.88	7201.76	0	1
C105	260.43	0.04	260.43	260.43	0.04	0	1
C106	284.87	7201.49	294.85	294.85	7210.04	0	1
C107	215.99	0.09	244.62	244.62	1.73	0	17
C108	200.88	0.2	200.88	200.88	0.2	0	1
C109	199.98	0.79	199.98	199.98	0.79	0	1

The BAP algorithm is terminated with a time limit of 7200 seconds. Table 2 reports our computational results for solving TDGVRP using the proposed

BAP algorithm. Column 1 reports the name of instances. Column 2 reports the lower bound (LB) at the root node. Column 3 reports the root time. Column 4 reports the best LB (BLB) after branching. Column 5 reports the upper bound (UB) corresponding to the best integer solution. Column 6 reports the total Cpu time. Column 7 reports the optimality gap ($Gap = (UB - BLB) / BLB * 100\%$). Column 8 reports the number of explored branch-and-bound tree nodes (Nodes). From Table 2, we can know that the proposed BAP algorithm can solve 6 out of 9 instances effectively, which proves the effectiveness and efficiency of the proposed BAP algorithm.

5 Conclusion and future research

We studied a TDGVRP with the consideration of traffic congestion. In order to solve the TDGVRP, we propose an exact BAP algorithm, where the master problem and the pricing sub-problem are solved by a CG algorithm and a labeling algorithm, respectively. The CG algorithm embedded within branch-and-bound framework is used to obtain feasible integer solution. Computational results prove the effectiveness and efficiency of the proposed BAP algorithm. At last, the work is worthy of further study. We can verify the BAP algorithm on larger scale instances, and we can also design an acceleration strategy to improve the algorithm.

References

1. Liu, R., Tao, Y., Xie, X.: An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. Computers & Operations Research **101**, 250–262 (2019)
2. Zhu, L., Hu, D.: Study on the vehicle routing problem considering congestion and emission factors. International Journal of Production Research **57**(19), 6115–6129 (2019)
3. Çimen, M., Soysal, M.: Time-dependent green vehicle routing problem with stochastic vehicle speeds: An approximate dynamic programming algorithm. Transportation Research Part D: Transport and Environment **54**, 82–98 (2017)
4. Malandraki, C., Daskin, M.S.: Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. Transportation science **26**(3), 185–200 (1992)
5. Figliozzi, M.A.: The impacts of congestion on time-definitive urban freight distribution networks co2 emission levels: Results from a case study in portland, oregon. Transportation Research Part C: Emerging Technologies **19**(5), 766–778 (2011)
6. Qian, J., Eglese, R.: Finding least fuel emission paths in a network with time-varying speeds. Networks **63**(1), 96–106 (2014)
7. Desaulniers, G., Desrosiers, J., Solomon, M.M.: Column generation, vol. 5. Springer Science & Business Media (2006)
8. Dabia, S., Ropke, S., Van Woensel, T., De Kok, T.: Branch and price for the time-dependent vehicle routing problem with time windows. Transportation science **47**(3), 380–396 (2013)
9. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research **35**(2), 254–265 (1987)